



Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

#3/01
#23/01
#15/01

11046 U.S. PTO
09/975961



066404
10#1

Bescheinigung

Certificate

Attestation

Die angehefteten Unterla-
gen stimmen mit der
ursprünglich eingereichten
Fassung der auf dem näch-
sten Blatt bezeichneten
europäischen Patentanmel-
dung überein.

The attached documents
are exact copies of the
European patent application
described on the following
page, as originally filed.

Les documents fixés à
cette attestation sont
conformes à la version
initialement déposée de
la demande de brevet
européen spécifiée à la
page suivante.

Best Available Copy

Patentanmeldung Nr. Patent application No. Demande de brevet n°

00440281.4

**CERTIFIED COPY OF
PRIORITY DOCUMENT**

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

I.L.C. Hatten

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE, 22/11/00
LA HAYE, LE

This Page Blank (uspto)



Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.:
Demande n°: 00440281.4

Anmeldetag:
Date of filing:
Date de dépôt: 18/10/00

Anmelder:
Applicant(s):
Demandeur(s):
ALCATEL
75008 Paris
FRANCE

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:

A data packet switching node accommodating very high bit rate interfaces

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat:
State:
Pays:

Tag:
Date:
Date:

Aktenzeichen:
File no.
Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:

/

Am Anmeldetag benannte Vertragsstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE/TR
Etats contractants désignés lors du dépôt:

Bemerkungen:
Remarks:
Remarques:

This Page Blank (uspto)

A DATA PACKET SWITCHING NODE ACCOMODATING VERY HIGH BIT RATE INTERFACES.

The present invention relates to a data packet switching node to be
5 used in an asynchronous digital network.

A frame switching relay is described in U.S. Patent No. 5,237,564.
According to this patent, such a frame switching relay comprises n input
ports and n output ports each of them having an identical binary bit rate D .
The switching relay comprises a time base at a frequency that is integral
10 multiple of the binary rate D . From this time base and by means of
frequency dividers, numerous clock signals required for the various
functions of the frame switching relay are derived. As a consequence, the
internal implementation of the frame switching relay is determined by the
bit rate D of the input and output ports.

15 However, the implementation of a switching fabric with ports
supporting very high bit rates (i.e. 9.6 Gbps and above) is at the limit of
technological feasibility and as a consequence very expensive. Indeed even
if a packet switching relay has only to accommodate one port at 9,6 Gbps,
all other ports being used at lower bit rates, the frame switching relay must
20 be designed as if all ports were to accommodate a bit rate of 9,6 Gbps. A
further disadvantage is that the resource of the switching fabric are wasted
if the whole switching fabric is designed for very high bit rates while several
ports accommodate lower bit rates.

Therefore, the object of the present invention is to provide a
25 simplified implementation of a data packet switching node able to switch
very high bit rates (i.e bit rates higher than the bit rate for which the
switching node is designed).

This object, and others that appear below, are achieved by a data
packet switching node according to claim 1.

The present invention also concerns a method according to claim 5.

Further advantageous features of the invention are defined in the dependent claims.

5 An advantage of the present invention is to extend the capabilities of a usual data packet switching node without modifying the core of the switching matrix.

Another advantage of the present invention is that it allows to flexibly configure the input and output ports of a usual data packet
10 switching matrix according to the needs.

In a preferred embodiment of the method, the ports can be dynamically configured depending on the bit rate supported by the input interface.

The data packet switching node according to the present invention
15 can be used in an ATM switch, a frame relay switch, an IP router or in any other device combining ATM switching and IP routing.

Other characteristics and advantages of the invention will appear on reading the following description of a preferred implementation given by way of non-limiting illustrations, and from the accompanying drawings, in
20 which:

- Figure 1 shows an embodiment of a data packet switching node according to the present invention;
- Figure 2 shows an example of inverse multiplexing on an input interface (Ingress Line);
- 25 - Figure 3 represents an example of multiplexing on an output interface (Egress Line);
- Figure 4 shows a bloc diagram of a data packet switching node according to the invention;

- Figure 5 represents the contents of the buffer memory storing the segments of a packet received on a very high speed interface.

Figure 1 shows an embodiment of a data packet switching node 10 according to the present invention. Data packet switching node 10 comprises eight input ports IP1, ..., IP8 and eight output ports OP1, ..., OP8. All input and output ports are designed to support an identical bit rate B (e.g. B=2,4Gbps).

According to the present invention, data packet switching node 10 can accommodate one input interface II1 with a bit rate of $k*B$ (e.g. if $k=4$ $k*B=9,6Gbps$ i.e. interface OC192c) in that 4 input ports IP1, ..., IP4 are bundled together. Each of the remaining four input ports IP5, ..., IP8 accommodate an input interface II2, ..., II5 with a bit rate of B (i.e. interface OC48c).

Input interface II1 is connected over splitter 11 to input ports IP1, ..., IP4. Input ports IP5, ..., IP8 are directly connected to respectively input interfaces II2, ..., II5.

According to the present invention, data packet switching node 10 can accommodate one output interface OI1 with a bit rate of 9,6Gbps (interface OC192c) in that 4 output ports OP1, ..., OP4 are bundled together. Each of the remaining four output ports OP5, ..., OP8 accommodate an output interface OI2, ..., OI5 with a bit rate of 2,4Gbps (interface OC48c).

Output ports OP1, ..., OP4 are connected over multiplexer 12 to output interface OI1. Output ports OP5, ..., OP8 are directly connected to respectively output interfaces OI2, ..., OI5.

This configuration is chosen for sake of simplicity, any other configurations may be envisaged. A more general configuration being n input and output ports, k input interfaces each input interface being

associated a certain number of input ports, k' output interfaces each output interface being associated a certain number of output ports. The following inequalities should be fulfilled:

$$\sum_{i=1}^k (nip)_i \leq n \quad \sum_{i=1}^{k'} (nop)_i \leq n$$

5 where :

$(nip)_i$ is the number of input ports associated to the i th input interface,

$(nop)_i$ is the number of output ports associated to the i th output interface.

It is the role of splitter 11 to split data packets received on input interface II1 into segments of constant length and successively retransmit
10 them on one of input ports IP1 to IP4 with a bit rate four times lower than the one received on input interface II1. Preferably, segments are cyclically retransmitted on input ports IP1, ..., IP4.

As shown on figure 2, if a packet received on input interface II1 can be split in eleven segments a to k, segments number a, e and i are
15 transmitted on input port IP1, segments number b, f and j are transmitted on input port IP2, segments number c, g and k are transmitted on input port IP3 and segments number d and h are transmitted on input port IP4. This function is called "inverse multiplexing in Ingress Line", (Ingress line designing an input interface).

20 In this example, it is assumed that all packets have the same length, however, this invention is not restricted to the switching of fixed length packets arriving on input interfaces II1, ..., II5. Packets with variable length can be handled the same way in that the packets are cut in segments of identical length, the last segment of a packet being if necessary filled with
25 dummy bits.

It is the role of multiplexer 12 to multiplex segments received on output ports OP1, ..., OP4 so as to reconstruct a packet on output interface

O11 having a bit rates four times higher than the bit rate on output ports OP1, ... ,OP4. Segments are, preferably, cyclically read by multiplexer 12 on output ports OP1, ..., OP4 and retransmitted on output interface O11. This function called "multiplexing in Egress Line " (the term Egress line being an equivalent for output interface). Data packet switching node 10 is responsible for properly assigning the switched segments to the output ports OP1, ...OP4 to guaranty that the segments are multiplexed at multiplexer 12 in the correct order on output interface O11.

An example for this reconstruction mechanism is given in figure 3.

10 If a packet is split in eleven segments m to w, the switching node should assign the segments of the output ports the following way: segments m, q and u should be received on port OP4, segments n, r and v should be received on port OP1, segments o, s and w should be received on port OP2 and segments p and t should be received on output port OP3. The

15 mechanism provided at data packet switching node 10 to preserve the correct order of the packet will be described below.

Figure 4 illustrates a bloc diagram for a data packet switching node according to the invention. Data packet switching node comprises a clock 40, a transport plane TP and a control plane CP.

20 Transport plane TP comprises an input stage 41, a buffer memory 42, an output stage 43. Input stage 41 is connected to the input interfaces I11, ..., I1k and to buffer memory 42. Output stage 43 is connected to buffer memory 42 and to output interfaces O11, ..., O1k'. The n input ports IP1, ..., IPn and splitter 11 are also part of input stage 41. As well, the n output

25 ports n OP1, ..., OPn and multiplexer 12 are part of output stage 43.

The mechanism for splitting a packet into segments in input stage 41 has already been described by means of Fig. 2. As well, the mechanism for reconstructing a data packet in output stage 43 has already been described by means of Fig. 3.

Clock 41 gives the clock frequency for the data packet switching node.

Preferably, if there are n input ports, each segment is virtually divided in n equally long parts called words in the following, a clock period
5 corresponding to the time needed to write a word in an input queue.

If the clock is coded on five bits (0 to 31), the memory contains enough place for storing 32 segments.

The segments received on input ports IP_1, \dots, IP_n are synchronized so that the beginning of a segment received on input port IP_i is delayed by
10 one word (one clock period) compared to the beginning of the segment received on the previous input port $IP_{(i-1)}$.

Such an organization of the input queues enables a high parallel management in the data packet switch node. The buffer memory management will be described in the following.

15 At each clock period, it is the turn of the next input port to write an available segment waiting in an input queue in the buffer memory 42.

At clock period i , the available segment is stored in at location i of buffer memory 42.

Figure 5 represents the contents of the buffer memory storing the
20 segments of a packet comprising twelve segments a to l received on input interface II_1 .

For example, if input ports IP_1 to IP_4 are associated to input interface II_1 ,
segment a of the packet received at clock period i on port IP_1 is stored at
25 location i of buffer memory 42,
segment b received at clock period $i+1$ on port IP_2 is stored at location $(i+1) \text{ MOD}(n)$,
segment c received at clock period $i+2$ on port IP_3 is stored at location $(i+2) \text{ MOD}(n)$,

segment d received at clock period $i+3$ on port IP4 is stored at location $(i+3) \text{ MOD}(n)$,

Segments (non represented on figure 5) stored between location $(i+4) \text{ MOD}(n)$ and $(i+n-1) \text{ MOD}(n)$ are segments received on input ports IP5 to
 5 IPn.

segment e received at clock period $i+4$ on port IP1 is stored at location $(i+n) \text{ MOD}(n)$,

segment f received at clock period $i+5$ on port IP2 is stored at location $(i+n+1) \text{ MOD}(n)$,

10 segment g received at clock period $i+6$ on input port IP3 is stored at location $(i+n+2) \text{ MOD}(n)$,

segment h received at clock period $i+7$ on port IP4 is stored at location $(i+n+3) \text{ MOD}(n)$,

Segments (non represented on figure 5) stored between location $(i+n+4) \text{ MOD}(n)$ and $(i+2*n-1) \text{ MOD}(n)$ are segments received on input ports IP5 to
 15 IPn.

segment i received at clock period $i+8$ on port IP1 is stored at location $(i+2n) \text{ MOD}(n)$,

segment j received at clock period $i+9$ on port IP2 is stored at location
 20 $(i+2n+1) \text{ MOD}(n)$ and so on.

This way of storing the segments in buffer memory 42 enables it to easily retrieve an implicit link between all segments of a packet.

In the following, figure 4 is further described.

The control plane CP comprises a translation table 45 and a traffic
 25 management module 46 comprising k' control queues (each one associated to an output interface) 461, ..., 46k'.

Translation table 45 contains routing information that is to say to which output interface(s) the packets arriving on an input interface should be switched to. Translation table 45 is preferably able to control the

switching for several virtual connections on one input/output port simultaneously.

The contents of translation table 45 determines the type of switching that is to be performed. Possible alternatives are point to point
5 switching, point to multipoint switching or multi point to point switching.

In this example, translation table 45 indicates that input interface I1 should be switched to output interface O1. It is, however, not necessary to switch an input interface with a bit rate $k \cdot B$ to an output interface with the same bit rate. Other routing combinations known by a person skilled in the
10 art are also supported by the present invention.

Translation table 45 comprises also a mapping, stored in a first memory location, between the input ports IP1, ..., IPn and their corresponding input interface I1, ..., Ik. As well translation table comprises a mapping, stored in a second memory location, between the output ports
15 OP1, ..., OPn and their corresponding output interface O1, ..., Ok'.

Traffic management module 46 is responsible for controlling the retransmission of a packet stored in the memory buffer 42 on an output interface O1, ..., Ok'. Traffic management module 46 controls the provision of quality of service requirements for the different packets. To
20 each output interface are associated as many control queues as available quality of services. In this example and for sake of simplicity, it will be assumed that all packets require the same quality of service and as a consequence only one control queue 461, ..., 46k' is associated to one output interface O1, ..., Ok'.

25 Each time a new packet is completely stored in buffer memory 42, a new entry is added in the control queue to which this packet should be switched according to translation table 45. Each entry in the control queue 46i indicates to output stage 43 the location in buffer memory 42 of the first segment of a packet to be retransmitted on output interface Oli.

Translation table 45 comprises as well the number of consecutive segments belonging to the same packet to be switched on output interface OP1.

The control queues 461, ..., 46k' are successively and cyclically
5 checked by output stage 43. Depending on the number of output ports associated to an output interface, the corresponding control queue will be checked by output stage 43 during the same number of clock periods. In the preceding example output interface OI1 corresponds to four output ports OP1, ..., OP4, then the control queue 461 will be checked for four
10 clock periods before control queue 462 is checked for one clock period and so on.

Control queues 461, ..., 46k' are handled with the FIFO (First In First Out) principle. At each clock period the next control queue is checked by output stage 43. If the transmission of a packet has been started and not
15 completed, it is the task of output stage 43 to retrieve in buffer memory 42 the subsequent segment of this packet. This mechanism will be described in the following. No entry is read out of the control queue and output stage 43 jump to the next control queue for the next clock period.

Else, if no packet is currently being transmitted, output stage 43
20 checks the control queue and read the address of the first segment of a new packet to be retransmitted of buffer memory 42.

It is also the role of output stage 43 to select the appropriate output ports OP1, ..., OP4 associated to output interface OI1 to retransmit the segments read out of buffer memory 42. Preferably, the first segment is
25 assigned randomly to one of the output ports associated to the output interface. Each subsequent segment is assigned to the next output port associated to the output interface. Alternatively, the output put is determined automatically from the value of the clock.

The mechanism used by output stage 43 to find the address of subsequent segments once the first segment has been read in the control queue will now be described. The address in buffer memory 42 of the next segment to be transmitted on an output port should be retrieved according to the following algorithm. Several conditions should be checked:

If the previous segment retransmitted on this output port was not the last one of the packet to be retransmitted on this port, the address of the next segment is given by the following table.

	Output interface associated to one output port	Output interface associated to $k > 1$ output port
Input interface associated to one input port	Address of previous segment sent on this port $+n$	Address of previous segment sent on this port $+k*n$
Input interface associated to $k > 1$ input ports	Address of the previous segment on this port $+1$ (if previous segment sent on this port was received on input port 1 to $k-1$)	Address of the previous segment sent on this port $+n$
	Address of the previous segment on this port $+ n-k+1$ (if previous segment sent on this port was received on input port k)	

If the previous segment retransmitted on this port was the last one of a packet to be transmitted on this port, two sub-cases should be considered:

If there is a new packet being currently transmitted on the output interface and one of the segment of this packet will be transmitted on this port, then the address of the next segment is given by the following table:

	Output interface 461 associated to one output port	Output interface 461 associated to $k > 1$ output ports
Input interface associated to one input port	Address of previous segment retransmitted on this port + n	Address of previous segment retransmitted on this port + n
Input interface associated to $k > 1$ input ports	If previous segment received on input port 1 to $k-1$	Address of the previous segment on this port + 1
	If previous segment received on input port k	Address of the previous segment on this port + $n-k+1$

- 5 If there is no new packet currently being retransmitted on the output interface to which this port is associated, the address of the segment to be retransmitted on this port is to be read in the control queue 461. It is the address of the first segment of a new packet).

- As already described by means of figure 3, the segments received
10 on the ports associated to output interface O11 are then combined to reconstitute the original packet.

- In a preferred embodiment of the invention, the association input ports/input interface, output ports/output interface as well as the number of input interfaces, resp. output interfaces, should be dynamically configurable
15 according to the needs of the data packet switching node.

This Page Blank (uspic,

CLAIMS

1/ A data packet switching node to be used in an asynchronous digital network, comprising:

- 5 - an input stage (41), cutting data packets into segments of constant length,
 - a switching matrix (42, 45, 46) for switching, said switching matrix having input ports (IP1, ..., IPn) and output ports (OP1, ..., OPn) supporting identical bit rates B;
 - and an output stage (43) reconstructing said data packets from said
 - 10 segments supplied by said output ports (OP1, ..., OPn) of said switching matrix,
- characterized in that
- said input stage (41) comprises at least one input interface (II1, ..., IIk) with a bit rate equal to a multiple of B, $k_i \cdot B$, and means (11) for splitting data
 - 15 packets received on said interface into segments distributed to k_i input ports of said switching matrix;
 - said output stage (43) comprises at least one output interface (OI1, ..., OIk') with a bit rate equal to a multiple of B, $k_o \cdot B$, and means for reconstructing a data packet with a bit rate equal to $k_o \cdot B$ by concatenating
 - 20 segments supplied by k_o output ports of said switching matrix; and
 - $k_i \cdot k_o > 1$.

2/ A data packet switching node according to claim 1, characterized in that said switching matrix comprises:

- a first memory location for storing an identifier representing the association
5 between said input interface and said corresponding ki input ports;
- a second memory location for storing an identifier representing the association between said output interface and said corresponding ko output ports.

10 3/ A data packet switching node according to claim 1, characterized in that said switching matrix comprises

- a buffer memory (42) for storing segments belonging to a packet received at said input interface (I1),
- memory writing means for sequentially writing segments received on said
15 ki input ports (IP1, ..., IP4) in said buffer memory (42);
- a translation table (45) for determining the output interface (OI1) to which said segments belonging to said packet must be switched;
- a traffic management module (46) for storing the address of the first segment of said packet in said buffer memory (42),
- 20 - memory reading means for retrieving consecutive segments belonging to said packet in said buffer memory and cyclically assigning each of said segments to one of said ko output ports (OP1, ..., OP4) associated to said output interface (OI1).

25 4/ A data packet switching node according to claim 1, characterized in that it is dedicated to be used in an ATM switch to switch fixed length data packets supplied on said input interface.

5/ A data packet switching node according to claim 1, characterized in that it is dedicated to be used in an IP router to switch variable length data packets supplied on said input interface.

- 5 6/ A data packet switching node according to claim 1, characterized in that it is dedicated to be used in an equipment providing both IP routing and ATM switching functions.

- 7/ A data packet switching node according to claim 2, characterized in that
10 the association between each input interface and corresponding input ports, as well as the association between each output interface and corresponding output ports are dynamically configurable in said first and second memory location.

This Page Blank (uspto)

A B S T R A C T**A DATA PACKET SWITCHING NODE ACCOMODATING VERY HIGH BIT RATE INTERFACES.**

The invention relates to a data packet switching node to be used in an asynchronous digital network. This data packet switching node comprises:

- an input stage, cutting data packets into segments of constant length,
- a switching matrix having input ports and output ports supporting identical bit rates B switching the segments;
- and an output stage reconstructing the data packets from the segments supplied by the output ports of said switching matrix,

According to the invention, the input stage comprises at least one input interface with a bit rate equal to a multiple of B , $k_i \cdot B$, and means for splitting the data packet into k_i input ports of the switching matrix. Moreover, the output stage comprises at least one output interface with a bit rate equal to a multiple of B , $k_o \cdot B$, and means for reconstructing a data packet with a bit rate equal to $k_o \cdot B$ by concatenating segments supplied by k_o output ports of said switching matrix where $k_i \cdot k_o > 1$.

FIG. 4

This Page Blank (uspic)

1/2

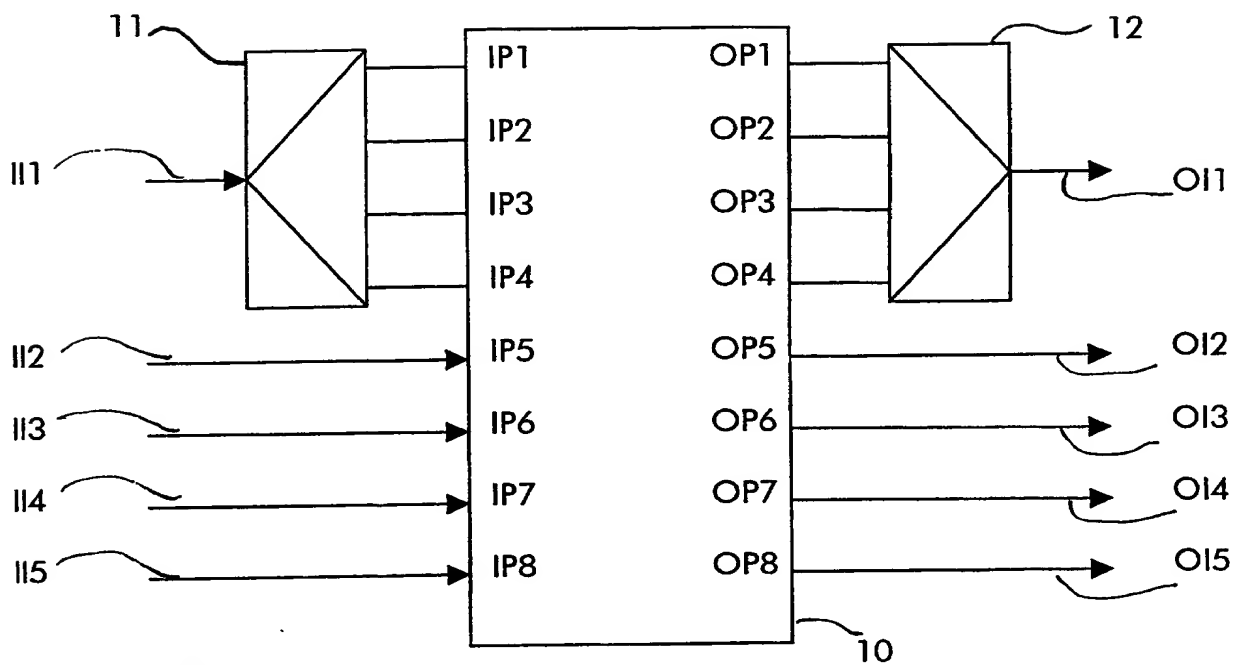


Fig 1

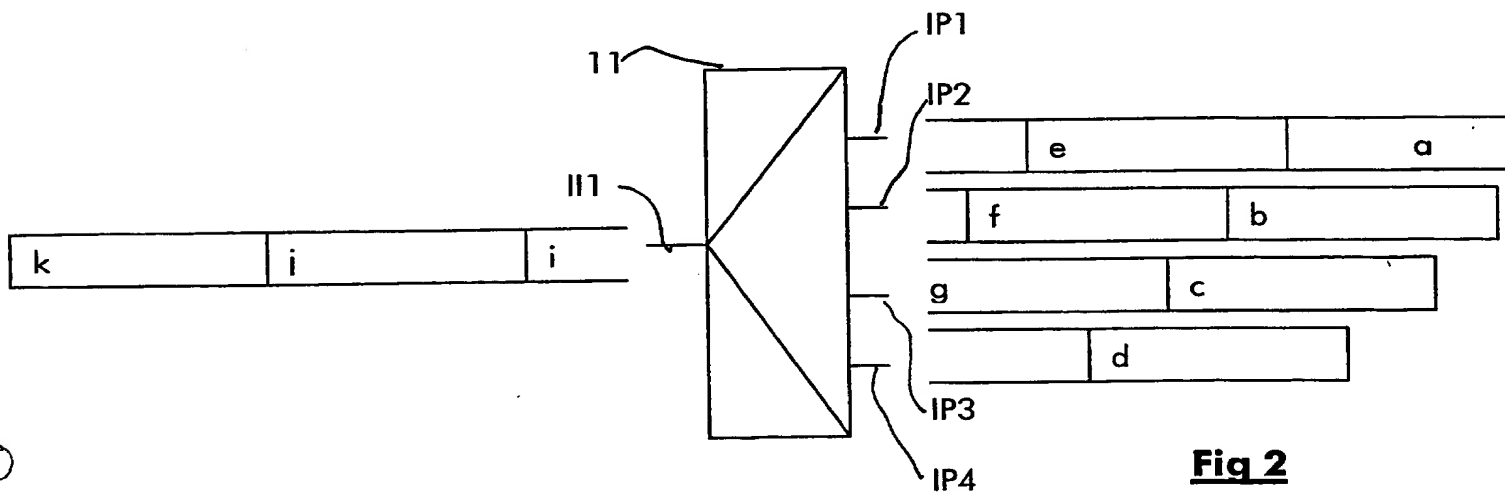


Fig 2

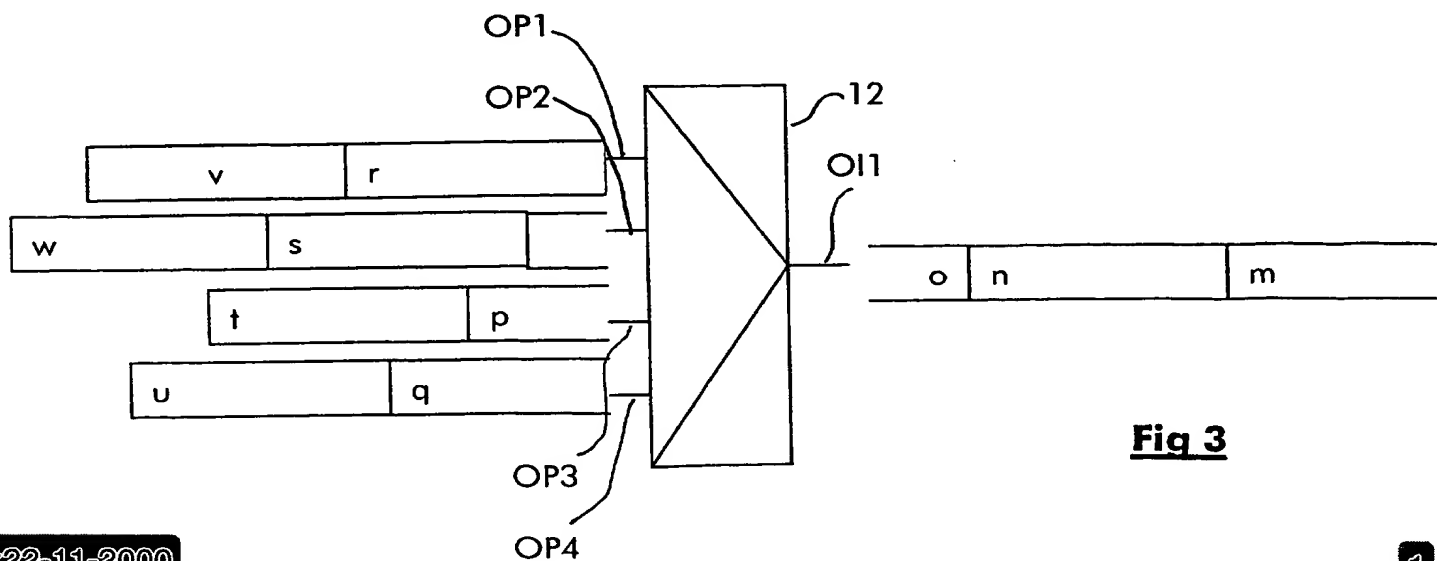


Fig 3

2/2

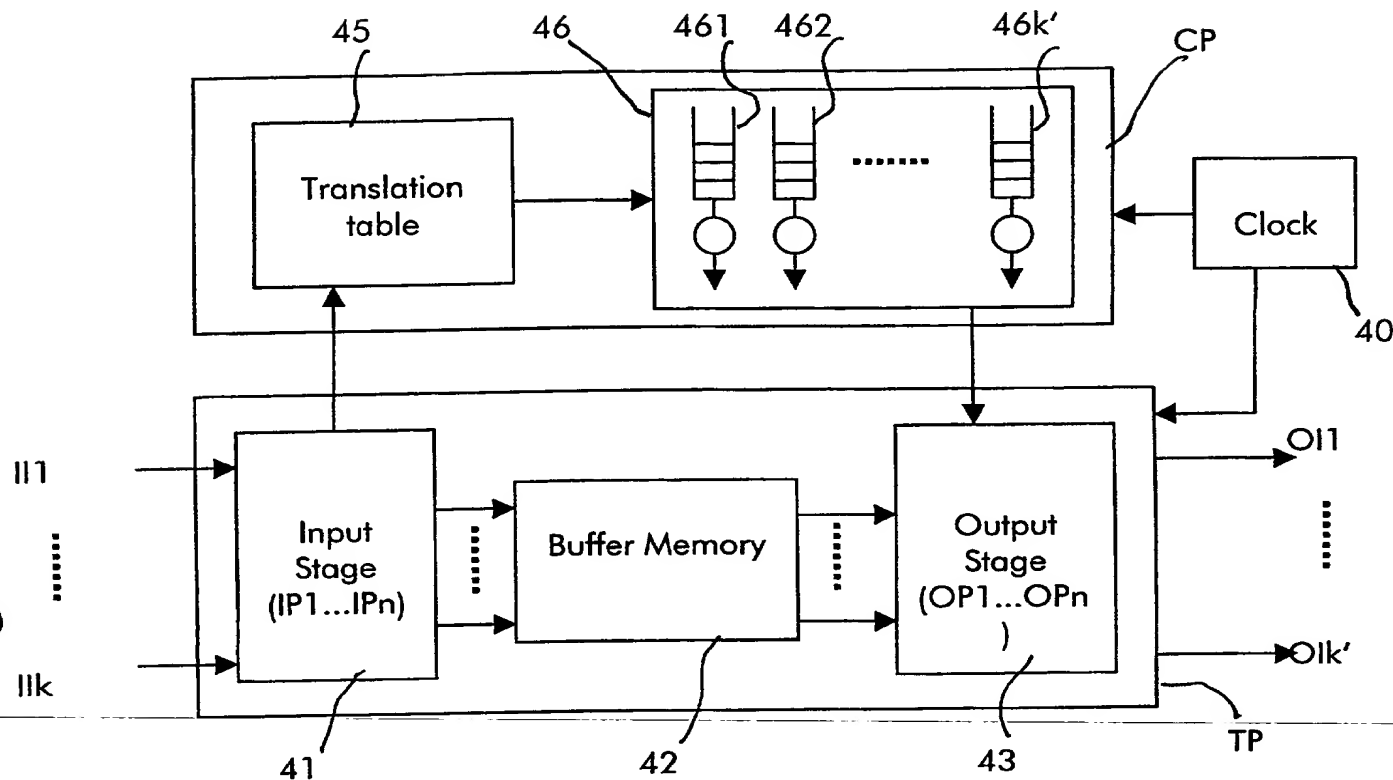


Fig 4

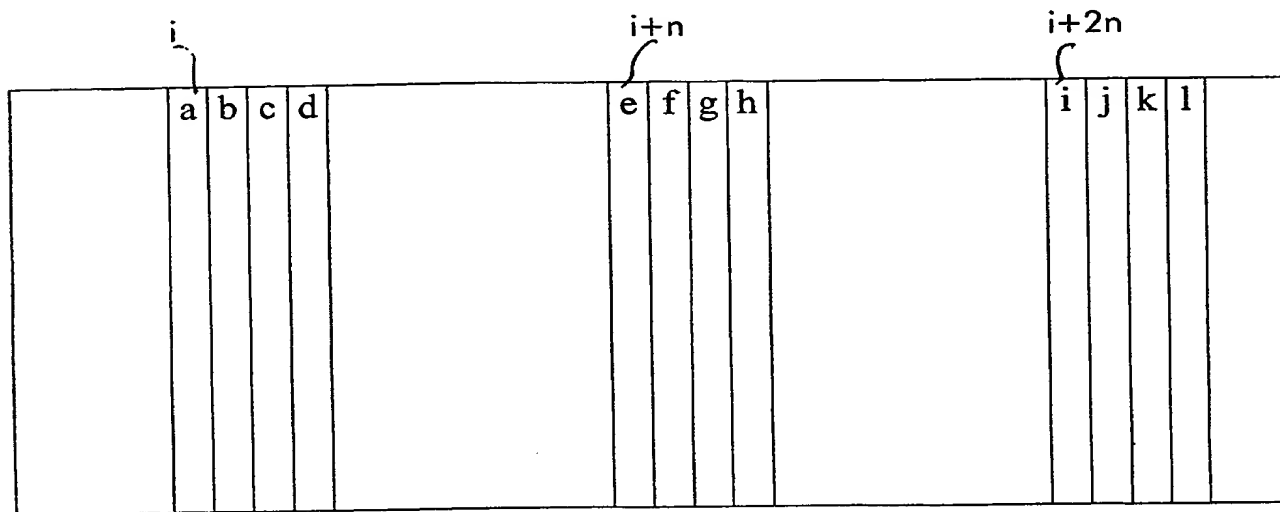


Fig 5

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

This Page Blank (uspto)